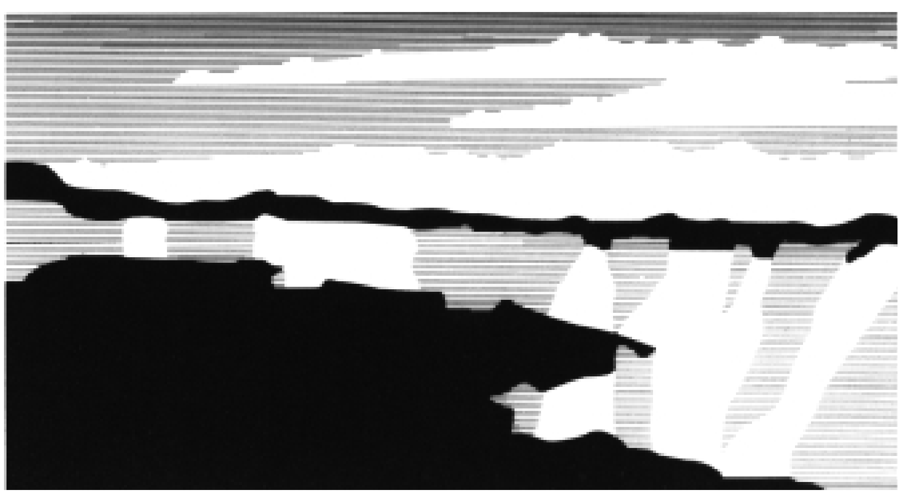
LA-UR-00-2578

Title: QWIC-2D DWM v1.0 User's Guide

Author: ***Eric R. Pardyjak and Michael J. Brown***
Energy and Environmental Analysis (TSA-4)

Submitted to: ***General Distribution***
May 2000



Los Alamos
NATIONAL LABORATORY

Form No. 836 R5
ST 2629 10/91

QWIC-2D DWM v1.0 User's Guide

DRAFT

May 31, 2000

Eric R. Pardyjak and Michael Brown
Los Alamos National Laboratory
Energy and Environmental Analysis Group, TSA-4

1.0 Introduction

As part of the Chemical Biological Non-Proliferation (CBNP) program, the Los Alamos National Laboratory has been working on fast response modeling in urban areas. These models enable information on the transport and dispersion of agents to be obtained quickly. To this end, a mass consistent diagnostic wind model for fast response in an urban environment is being developed.

This document describes the 2D version of the code QWIC-2D-DWM (Quick Wind Interpolation – 2D – Diagnostic Wind Model) a simple diagnostic wind model (written in FORTRAN77). QWIC-2D-DWM allows the user to input randomly spaced wind data (in time and space) from various measuring stations in a given domain. Using an interpolation algorithm and a mass conservation scheme, a 2D gridded mass consistent wind field is obtained.

In the near future, a second version, QWIC-3D-URB, will be developed that will produce wind fields within various 3D building configurations. It will be derived from the QWIC-2D-DWM with additional algorithms to describe flow around buildings. This user's guide only describes QWIC-2D-DWM.

2.0 Model Theory

In this section, the theory behind the model is briefly described. The QWIC-2D model developed for the CBNP is based on the mass consistent model of Sherman (1978) and the implementation of Kaplan and Dinar (1996). The 2D-interpolation algorithm is based on the Barnes Objective Map Analysis Scheme as developed by Koch et al. (1978). Below an overview is given of the interpolation scheme and the mass consistency scheme. The theory section will be expanded in a future draft.

2.1 Barnes Interpolation Scheme

The Barnes scheme used in QWIC-2D allows for randomly spaced wind data to be interpolated onto a uniform grid whose size and dimension are specified by the user. The Barnes (1973) scheme that was implemented allows for two passes over the observed data. The first pass is an interpolation pass based on a simple Gaussian-weighted averaging technique and the second pass is a so-called "correction pass". The second pass effectively iterates on the first set of gridded data, compares it to the original set of measurements, and then adjusts the gridded data (using a user-defined convergence parameter) to minimize the difference between the interpolated data and the observations. At this point, the gridded interpolated wind field does not, in general, satisfy mass conservation.

2.2 The Mass Consistent Diagnostic Wind Model

The mass consistent scheme used in QWIC-2D was originally developed by Sherman (1978). This model was chosen over a potential flow model because of the model's ability to allow for rotational flow. A mass consistent wind field is obtained using the method of Lagrange multipliers. With this method, the difference between the gridded interpolated wind field and the mass consistent velocity field is minimized.

3.0 Running the Code

In this section the components of the QWIC-2D-DWM code and how to compile the code using a makefile are described. A description of the input and output files is given along with suggestions for modifying input parameters.

3.1 The Makefile

The QWIC-2D code consists of the following five subroutines:

```
div.f        interp.f     velup.f
main.f       sorsu.f      sort.f
```

a makefile and an input file called `input.dat`.

The main program is `main.f`. The program is compiled using a makefile. Figure 1 is the makefile that has been tested on the following UNIX workstations: SGI, HP and SUN.

```
# makefile for main.f for QWIC-2D-DWM, below is the link and
# compile section

# link section ====================================
qwic2d.x: main.o sorsu.o div.o velup.o interp.o
        f77 main.o sorsu.o div.o velup.o interp.o -o qwic2d.x

#compile section =========================
main.o: main.f
        f77 -c main.f
sorsu.o: sorsu.f
        f77 -c sorsu.f
div.o: div.f
        f77 -c div.f
velup.o: velup.f
        f77 -c velup.f
interp.o: interp.f
        f77 -c interp.f
```

Figure 1 – The makefile for QWIC-2D-DWM.

The makefile is run by simply typing “make” on the UNIX command line in the directory of the makefile and FORTRAN routines. The name of the executable routine is qwic2d.x. With an appropriate input file in place, the code is run by typing “qwic2d.x”. The code produces output to the screen as shown below in Figure 2.

```
Running QWIC-2D-DWM for salt_lake_city
  Domain and Grid Size DWM:
Lx =      50.00 Ly =      150.00
dx =       1.00 dy =        1.00
 Writing Data to files:
 velcol.dat
 velblk.dat
 qwic2d.chk
```

Figure 2 - Screen output display while running QWIC-2D.

3.2 The Input File

Running QWIC-2D requires one input file. The input file is shown below in Figure 3.

```
Salt_lake_city        !location
03151999              !date (mmddyyyy)
2312                  !time (hhmm)
50.                   !Lx – domain length x-direction(E-W) in km
150.                  !Ly – domain length y-direction(N-S) in km
51                    !Nx – no. grid points in x-direction(E-W)
151                   !Ny - no. grid points in y-direction(N-S)
.01                   !alpha – error parameter
1.78                  !w - SOR relaxation factor
1.0                   !ffact - interpolation forcing factor
400.                  !utmxsw - western edge of domain (km)
4425.                 !utmysw - southern edge of domain (km)
13                    !number of sites listed below
  utmx(m)               utmy(m)         u(m/s)    v(m/s)
415530.0758           4507046.1218       1.64     -0.44
438738.3501           4452424.2744       2.30     -1.93
425867.1866           4526919.0091       1.44     -1.39
428125.6265           4499142.3351       2.65      3.39
432142.0400           4475792.2149       5.26     -3.82
426533.3412           4509149.3518       2.62      1.45
439695.9349           4465737.0611       3.56      0.50
407927.6070           4507136.5689      -2.06     -0.40
443868.3171           4455714.4525      -0.43     -2.46
418150.0965           4514788.1915       4.70     -1.71
428029.1763           4489152.0132      -0.21      1.69
418981.7769           4513668.8332       0.89      0.14
```

Figure 3 - Input data file – input.dat – for QWIC-2D.

The first three lines are used for output file identification purposes and do not impact the diagnostic model simulation. The first line of `input.dat` is a city name indicator. The second line is an 8-digit date in a month, day, year format (mmddyyyy). The third line is a four-digit 24-hour time code in an hour and minute format (hhmm). The fourth and fifth lines, `Lx` and `Ly` are the domain lengths in kilometers in the east-west (x) and north-south (y) directions. `Nx` and `Ny` are the number of grid points in the east-west (x) and north-south (y) directions. The maximum number of grid points allowed in each direction is 201, but this can be changed within the Fortran code. The domain lengths `Lx` and `Ly` and the number of grid points `Nx` and `Ny` should be chosen such that the east-west (dx) and north-south (dy) grid spacings are equal. Alpha (α) is the Gaussian precision modulus and is a function of the observation error (α is discussed more thoroughly below in Section 3.4. The successive over-relaxation (SOR) over-relaxation factor (ω) is usually taken to be 1.78 based on the recommendations of Sherman (1978). The `ffact` variable is used to modify the effective response wavelength in the Gaussian interpolation scheme (the code is quite sensitive to this parameter). Lines eleven and twelve contain the Universal Transverse Mercator (UTM) coordinates of the southwest corner of the computational domain in kilometers. Line thirteen is the number of observation sites to be used in the diagnostic analysis. The number of observation sites is restricted to 50 or less (unless the parameter `maxdat` is changed in the `interp.f` subroutine). The remaining rows contain four columns of data. The first two columns are the UTMX and UTMY coordinates of the observation sites (in meters) and the last two columns are the observed westerly (u) and southerly (v) components of the wind velocities (in meters/second) at these sites for a particular time. A current beta version of QWIC-2D also accepts time-dependent meteorological data. The wind data needs to be manually input or “cut and pasted” into the `input.dat` file. A specialized preprocessing code has been written for reading in meteorological data files from the Utah Mesonet, converting the data from wind velocity and direction to wind components, and outputting the data in the four column format described above.

3.3 Output Data Files

The users can modify the source code to produce any output they want, however the standard version of QWIC-2D has the following output files:

`velcol.dat`　　`velblk.dat`　　`qwic2d.chk`

All output files are in ASCII format. The `velcol.dat` file contains the interpolated and mass conserved velocity field data in column output form. The FORTRAN format for `velcol.dat` is: `2(2x,f8.2),2(2x,f8.5)`, where the first two columns are the UTMX and UTMY coordinates and the last two columns are the westerly (u) and southerly (v) wind velocity components. The `velblk.dat` output file contains only the westerly and southerly wind velocity components (without the location coordinates) in a block format (`20f5.1`) accepted by the EPA’s dispersion code INPUFF. Figure 4 shows a sample calculation of the gridded mass consistent data overlaid onto the observed data for Salt Lake City, UT using the `velcol.dat` file.

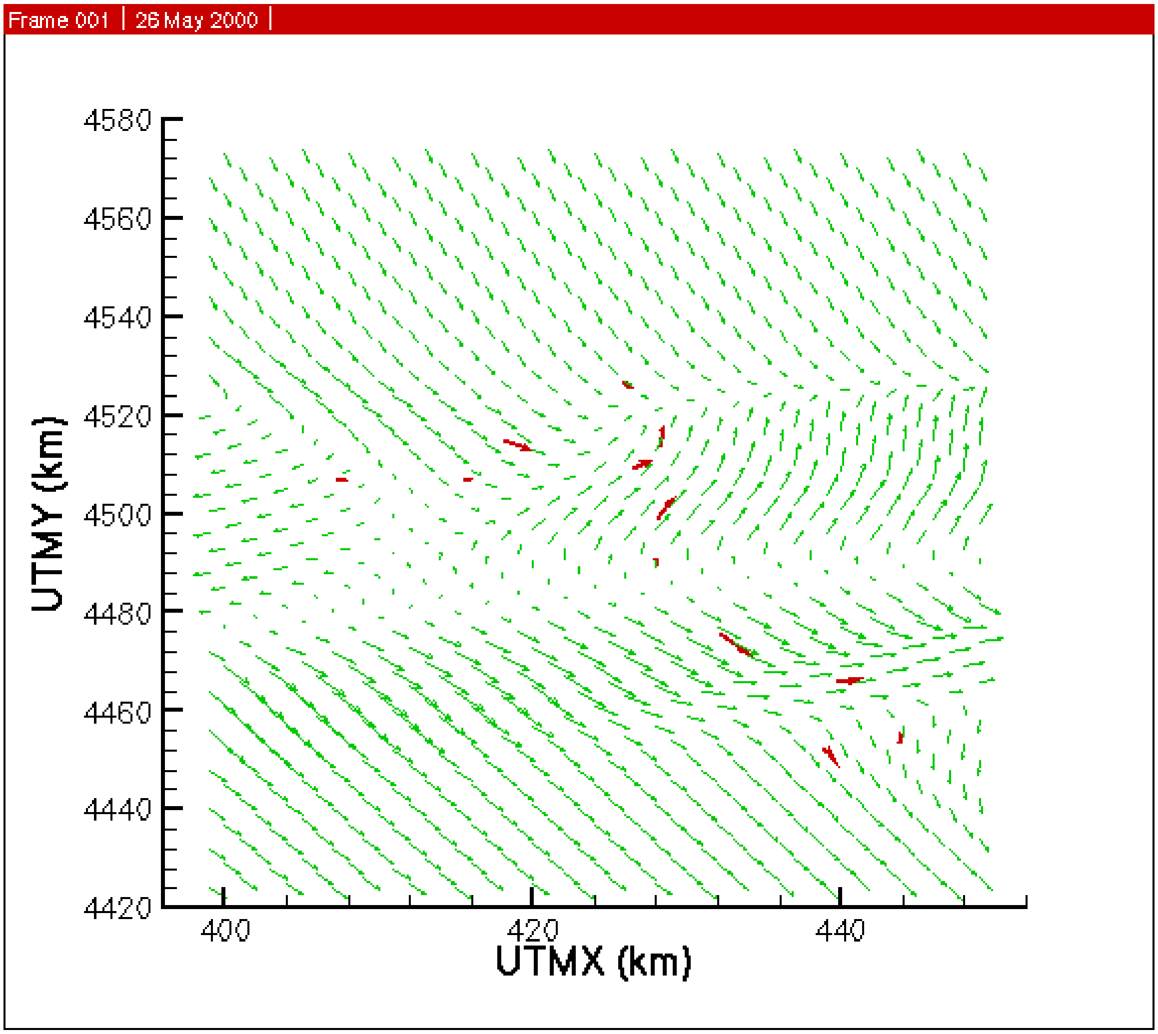


Figure 4 - Sample output from QWIC-2D (green vectors) compared with observed data (thick red vectors).

The file `qwic2d.chk` contains the data from the input file to help verify that the appropriate data were entered in `input.dat`. Note that the UTMX and UTMY coordinates are written out in units of kilometers. In addition, `qwic2d.chk` contains the model-produced interpolated data and the observed data at each observation location. The model-produced data are obtained by bilinearly interpolating to the observed locations using the gridded data set. Figure 5 is a sample `qwic2d.chk` file. The last two lines of the file are the average norm differences between the measured data and the interpolated data at the observation locations.

```
Salt_lake_city       !location
03151999             !date (mmddyyyy)
2312                 !time (hhmm)
50.                  !Lx – domain length x-direction(E-W) in km
150.                 !Ly – domain length y-direction(N-S) in km
.01                  !alpha – error parameter
1.78                 !w - SOR relaxation factor
1.0                  !ffact - interpolation forcing factor
400.                 !utmxsw - western edge of domain (km)
4425.                !utmysw - southern edge of domain (km)
151.                 !number of sites listed below
utmx                  utmy               u(m/s)    v(m/s)
415530.0758          4507046.1218       1.64     -0.44
438738.3501          4452424.2744       2.30     -1.93
425867.1866          4526919.0091       1.44     -1.39
428125.6265          4499142.3351       2.65      3.39
432142.0400          4475792.2149       5.26     -3.82
426533.3412          4509149.3518       2.62      1.45
439695.9349          4465737.0611       3.56      0.50
407927.6070          4507136.5689      -2.06     -0.40
443868.3171          4455714.4525      -0.43     -2.46
418150.0965          4514788.1915       4.70     -1.71
428029.1763          4489152.0132      -0.21      1.69
418981.7769          4513668.8332       0.89      0.14
428265.2580          4513573.0697       0.54      3.86

interpolated u-velocity absolute norm: 0.4567371368155139
interpolated v-velocity absolute norm: 0.2500982044511596
```

Figure 5 - Sample qwic2d.chk file

3.4 Choosing the Input Parameters

As described in section 1.3, there are various parameters that need to be entered into the `input.dat` file. The domain lengths, Lx and Ly, currently must be chosen such that the resulting grid size is uniform. That is, `dx=dy`, where `dx=Lx/(Nx-1)` and `dy=Ly/(Ny-1)` and where `Nx` and `Ny` are the number of grid points in the domain in the x (east/west) and y (north/south) directions, respectively. The maximum array size for `Nx` and `Ny` are set to 201 for both variables. This can be increased by changing the variables `nxx` and `nyy` in the `main.f` program.

The variable `alpha` determines the strength with which the code forces the correction of the gridded interpolated wind field to satisfy mass conservation (Kaplan and Dinar, 1996). A larger `alpha` results in a stronger mass conservation requirement, but may result in winds that do not match the observations as well. According to Sherman (1978) the value of `alpha` should be chosen as function of the fluctuation velocity component (i.e., α^2=0.5σ^2). For the two-dimensional cases tested, the final calculated velocities are not extremely sensitive to `alpha's` value.

The interpolated velocity field is very sensitive to the variable `ffact`, a non-dimensional scaling factor. Changing `ffact` changes the shape of the Gaussian filter being used for the weighting function and the effective length scale of the interpolation. In general, `ffact` should be close to unity to satisfy sampling theory criteria built into the algorithm.

4.0 Description of the Program and Subroutines

QWIC-2D contains a main program and five subroutines. Within the source code itself, there is a brief description of what each subroutine does along with comments within the code. Below is a brief description of the main program and each of the subroutines.

4.1 Program main.f

Program `main.f` is fairly short, but performs several tasks. It calls all of the subroutines (`interp.f`, `div.f`, `sorsu.f`, and `velup.f`) which perform the interpolation and mass consistent diagnostic wind calculations and it opens and closes all of the input and output files. Several variables are read from `input.dat` in `main.f`. The output files `velcol.dat` and `velblk.dat` are written out here and parts of `qwic2d.chk` are also written out in `main.f`. The Lagrange multiplier matrices are initialized in `main.f` for inflow boundary conditions only (i.e., no walls).

4.2 Subroutine interp.f

The subroutine `interp.f` uses the Barnes Objective Map Analysis Scheme (BOMAS) as implemented by Koch et al. (1983) to calculated the initial gridded velocity field that is used by the mass consistent diagnostic wind model. Within `interp.f`, the rest of the `input.dat` is read into memory including the observation data. The average distance between each of the measuring sites is calculated and used for determining a characteristic length scale for the BOMAS. A first interpolation pass through the grid is made using a Gaussian filter width based on the average distance between the wind observation locations. The gridded velocity data is then interpolated onto the observation data sites using a simple bilinear interpolation scheme. The interpolated data is compared to the observed data and a second pass on the grid is made (using a smaller filter width) to reduce the difference between the interpolated velocity data and observed velocity data.

It should be noted that the interpolated velocity data (`uo` and `vo`) are on a grid of size `nx+2` by `ny+2`, while mass consistent velocity data are on a grid of `nx` by `ny`. This is done to allow for ease of implementation of the differencing scheme.

4.3 Subroutine div.f

This subroutine simply calculates the right hand side of the Poisson's equation found in the mass consistent diagnostic wind model framework (Sherman, 1978). The divergence of the interpolated velocity field is calculated using a central difference stencil.

4.4 Subroutine sorsu.f

This subroutine is the Successive Over-Relaxation (SOR) solving subroutine. In this subroutine, the Poisson's equation for the Lagrange multipliers is solved for using a central difference stencil by iterating until the point-wise error is reduced below 1e-6 or until 10,000 iterations are made.

4.5 Subroutine velup.f

Subroutine `velup.f` calculates the mass consistent wind field based on the iterated Lagrange multipliers calculated in subroutine `sorsu.f` and the original interpolated velocity field.

5.0 Future Work – Model improvements

The current model will be expanded to 3D. In addition, features that improve the model's ability to predict flow patterns will added. This will include urban flow effects. That is, empirical models of the large-scale flow structure that are influenced by buildings will be added. In addition, more physics will be added to the model to account for terrain and stability effects.

Any comments or corrections to the code or this document would be appreciated. Please send any comments or corrections to: pardyjak@lanl.gov

6.0 Acknowledgments

This project was funded by the Department of Energy as part of the Chemical Biological Non-Proliferation (CBNP) program under the direction of Page Stoutland (page.stoutland@hq.doe.gov).

7.0 References

Kaplan, H. and N. Dinar. A Lagrangian Dispersion Model for Calculating Concentration Distribution within a built-up domain, *Atmos. Env*., Vol. 30, No. 24, pp. 4197-4207, 1996.

Dinar, N., Mass Consistent Models for Wind Distribution in Complex Terrain – Fast Algorithms for Three Dimensional Problems, *Boundary-Layer Met.* Vol. 30, pp. 177-199 1984.

Sherman, C. A., A Mass Consistent Model for Wind Fields over Complex Terrain, Journal of Applied Meteorology, vol. 17, 1978.

Koch, K.E., M. DesJardins, P.J. Kocin, An Interactive Barnes Objective Scheme for Use with Satellite and Conventional Data, *Journal of Climate and Applied Meteorology*, vol.22, no. 9, pp. 1487-1503.